



[illegible][illegible][illegible][illegible][illegible]









0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

[0019] An arbiter may be used with a processor in a point-to-point architecture to allow a customer to change the number of ports linked between a processor and a device external to the processor. A point-to-point bus architecture provides an uninterrupted connection between two separate devices. Generally, a packet based protocol transfers information in a point-to-point bus architecture. The arbiter may alter the number of ports linked between the processor and the device exterior to the processor by changing one or more internal signal pathways in the processor without changing a physical component layout in the processor. In an embodiment, the manufacturer or the customer may change the number of ports linked between the two devices to increase the bandwidth between the devices. In an embodiment, the manufacturer or the customer may add additional processors linked to a device by changing the number of ports linked between a first processor and the device exterior to the first processor.

042390.P10139

processor **200** has an arbiter **202** and three layers, the protocol layer **204**, a buffer layer **206** such as one or more link layers **210**, and an information transfer layer **208** such as one or more physical layers **212**. The arbiter **202** changes the signal pathways within the physical layer **212** and link layer **210** without changing the physical component layout in the processor **200**. The physical layer **212** carries out the actual physical transfer of information to and from other devices. The link layer **210** performs liaison functions between the higher functions of the protocol layer **204** and the physical layer **212**. The protocol layer **204** processes requests, responses, and data transfers.

[0021] In an embodiment, the arbiter **202** may be internal or external to the processor **200**. In an embodiment, a single arbiter **202** controls the signal pathways in all the layers of the processor **200**. In an alternative embodiment, multiple arbiters **202** exist to control the signal pathways. The arbiter **202** may be a combination of hardware and software. The arbiter **202** may have several functions such as sending an enable/disable signal to one or more signal pathway switching devices and sending a signal to change various clocking speeds. The buffer layer **206** may have one or more inbound signal pathways. The buffer layer **206** may have one or more outbound signal pathways. The information transfer layer **208** may have one or more inbound signal pathways. The information transfer layer **208** may have one or more outbound signal pathways.

[0022] Figure 3 illustrates a two processor point-to-point architecture **300** having a 16-bit point-to-point connection between: 1) an input-output component **302** and the first processor **304**; as well as 2) the input-output component **302** and the second processor **306**. A first





must occur to transfer eight 16-bit packets (128 bits) of information between each processor and the input-output component. In 32-bit mode, four clock cycles must occur to transfer four 32-bit packets (128 bits) of information between each processor and the input-output component.

[0024] Figure 5 illustrates an embodiment of a four processor point-to-point architecture **500** having a 16-bit point-to-point connection between an input-output component **502** and each of the four processors **504, 506, 508, 510**. The four processors are processor 1 **504**, processor 2 **506**, processor 3 **508**, and processor 4 **510**. In an embodiment, each of the four processors **504, 506, 508, 510** employs an embodiment of the arbiter. The processors **504, 506, 508, 510** have a flexible architecture that coordinates with the arbiter to allow a variety of uses for these processors **504, 506, 508, 510**. For example, the four processor architecture **500** may be substituted with the two processor architecture employing a 32-bit connection with the input-output component **502**. Thus, the processing power of this arrangement has effectively doubled because four processors will process the data coming from the input-output component **502**. Thus, an embodiment of the arbiter allows the same input-output component **502** to work with either a two processor architecture, a four processor architecture **500** or other similar multiple processor architecture.

[0025] A programmable knob setting in a configuration register directs the arbiter to establish the customer's current desired configuration such as a 16-bit, 32-bit, or 48-bit point-to-point connection between the processor and a device exterior to the processor. In an embodiment, the device exterior to the processor may be another processor, an input-output device, a bridge or other similar device. In an embodiment, the manufacturer or the customer may program the knob

[0026] Typically, in a server, a large exchange of data occurs between a first processor and a second processor. Thus, having a 48-bit point-to-point connection between these two devices greatly decreases the time required to complete each transaction between the devices. Typically, in a work station, a large exchange of data occurs between a each processor and the input-output component. Thus, having a 32-bit or 48-bit point-to-point connection between these devices greatly decreases the time required to complete each transaction between the devices.

[0027] Figure 6 illustrates an embodiment of an eight processor (8P) point-to-point architecture 600 comprised of four pairs of processors, 602, 604, 606, 608, 610, 612, 614 and 616, linked to a corresponding bridge, 620, 622, 624, 628, and each bridge, 620, 622, 624, 628, connected to a chipset 630. A first processor 602 and a second processor 604 each have a 16-bit point to point connection with a first bridge 620. The first bridge 620 has a connection, such as a 32-bit connection, with the chipset 630. Similarly, a third processor 606 and a fourth processor 608 each have a 16-bit point to point connection with a second bridge 622. The second bridge 622 has a connection, such as a 32-bit connection, with the chipset 630. In a similar fashion, the fifth

processor **610** through eighth processor **612** eventually link with the chipset **630**. In an embodiment, the arbiter may allow the signal paths internal to each processor **602, 604, 606, 608, 610, 612, 614** and **616**, to be changed by a programmable setting. This allows a manufacturer to fabricate a single version of a processor with a generic but flexible architecture within the processor to service multiple processor platforms. Thus, in an embodiment, an arbiter linked to a processor having a single flexible architecture may be employed to service, a 2P architecture, 4P architecture, 8P architecture or other multiple processor architecture, as well as service a server application and a workstation application.

[0028] In an embodiment, the ability of the arbiter to change the allocation of internal signal paths allows a manufacturer to design and fabricate fewer versions of the same chip to service customers needs. In an embodiment, the architecture of the processor is the arrangement and component makeup of the various elements within the processor. In an embodiment, the components and elements contained within the protocol layer, the information layer and the buffer layer makeup the architecture of the processor. Those ordinarily skilled in the art will recognize the specific components illustrated in the following embodiments may be deviated from and still be within the spirit of this invention.

[0029] Figure 7 illustrates an embodiment of the arbiter **702** controlling the inbound signal pathways in an embodiment of the information transfer layer **700**. The first physical layer (SPXO) **704** has a first port **706**; nine 16-bit registers, a first register **708**, a second register **710**, a third register **712**, a fourth register **714**, a fifth register **716**, a sixth register **718**, a seventh register **720**, an eighth register **722**, and a ninth register **724**; first 64-bit register **726**; a signal pathway







[0033] In an embodiment of the inbound buffer layer **800**, if the configuration register **832** is programmed for a 32-bit point-to-point connection, then the arbiter **802** effectively gangs two link layers together. The first link layer **804** receives the eight related 16-bit packets (128-bit) of information at twice the clock speed from the communication switching device in the first physical layer. The first link layer **804** stores the 128-bit packet of information in the first register **806**. The 128-bit packet of information is routed appropriately to either the first response queue **808** or the first request queue **810**. The arbiter **802** sends an enablement signal to the first signal pathway switching device **812**, third signal pathway switching device **834**, and fifth signal pathway switching device **822**. The arbiter **802** also sends a disable signal to the second signal pathway switching device **816** and fourth signal pathway switching device **836**. The arbiter **802** directs the fifth signal pathway switching device **822** to request data only from the first signal pathway switching device **812** and the third signal pathway switching device **834**. In 32-bit mode, the second signal pathway switching device **816** and fourth signal pathway switching device **836** receive no signal from the physical layer because the arbiter **802** disabled the second flip flop and fourth flip flop in the physical layer.

[0034] The 128-bit packets of information are transferred to the fifth signal pathway switching device **822** through the first signal pathway switching device **812** and then the third signal pathway switching device **834** in a cyclic sequential manner. In a repeating cycle, the fifth signal pathway switching device **822** requests the contents of the first response queue **808**, then the first request queue **810**, then the third response queue **838**, then the third request queue **840**, and then repeats this sequence. Every packet of information may be either a request for a command and/or data or a response to a command or data. In an embodiment, once a packet is written into a





layer **840**.

[0036] The inbound link layer sends responses and requests to protocol layer **840**. The protocol layer **840** alternates choosing a command from the LATT **826** or RATT **830** accompanied by data from the CDM **828**. In one embodiment, this physical information may be sent to a CPU. The information goes to the CPU to be processed and sent to the outbound bound link layer. The protocol layer **840** typically is the processing component. The protocol layer **840** typically resolves requests from other local processors and sends requests out through the physical layer. The protocol layer **840** may be responsible for functions such as ordering, conflict detection, and coherence. The protocol layer **840** may forward requests for data to the local memory controller if the request is made to local memory. The protocol layer **840** may send snoop requests to the remote nodes and other local processors. The protocol layer **840** also processes requests made through the link layer, which may come from a remote processor or an input-output component.

[0037] Figure 9 illustrates an embodiment of the arbiter **902** controlling the outbound signal pathways in an embodiment of the buffer layer **900**. The protocol layer **904** sends the processed data and commands to each outbound link layer **906**, **908**, **910**, **912**. The protocol layer **904** delivers commands generated from itself to the LATT **914** and data, if any, corresponding to that command to the CDM **916**. The protocol layer **904** delivers responses to commands from remote devices to the RATT **918** and data, if any, corresponding to that command to the CDM **916**. If the configuration register **920** is programmed for a 32-bit point-to-point connection, then the arbiter **902** effectively enables two of the four outbound link layers. The following happens in

tracing the signal path in the first outbound link layer **906** and the second outbound link layer **908**. The arbiter **902** directs the second signal pathway switching device **922** and the fourth signal pathway switching device (not shown) to be disabled. The arbiter **902** directs the fifth signal pathway switching device **928** to request data through the first signal pathway switching device **924** and third signal pathway switching device **926** from their respective request buffer **930** or response buffer **932**. The fifth signal pathway switching device **928** alternates, in a repeating cycle, requests for bits from the request buffer **930** and the response buffer **932**. The fifth signal pathway switching device **928** also receives any data, if any, corresponding to those commands from the CDM **916**. The fifth signal pathway switching device **928** then routes the 128-bit data packet through a first time delay **933** to either the outbound response queue **936** or outbound request queue **934** located in the outbound physical layer.

**[0038]** One or more retry queues **938** exist in the outbound link layer. The retry queue **938** stores a duplication of the information to be transferred in case an error occurs in the transfer of the information. The retry queue **938** exists primarily in case there may be an error in the link. The retry queue **938** erases the duplicated information stored in a retry queue **938** upon receiving a confirmation of an error free transfer of information.

**[0039]** If the configuration register **920** is programmed for a 16-bit point-to-point connection, then the arbiter **902** directs each outbound link layer **906**, **908**, **910**, **912** to act independently. The arbiter **902** sends a signal to enable all of the outbound link layer communication switching devices and set the clock speed to the default clock speed. The information is routed from the LATT **914**, RATT **918**, and CDM **916** through its own respective outbound link layer to the corresponding outbound physical layer. For example, the arbiter **902** directs the second signal

[0040] Figure 10 illustrates an embodiment of the arbiter **1002** controlling the signal pathways in an outbound information transfer layer **1000**. In an embodiment, the information transfer layer comprises one or more physical layers. The physical layer carries out the actual physical transfer of information to devices external to the processor through ports, such as a first port **1004** and a second port **1006**. Typically, one 16-bit port exists on each physical layer such as the first port **1004**. The processor transfers information between itself and a device exterior to the processor through these ports. The information signal contains information such as a request, response, or data.

[0041] In an embodiment of a two processor architecture, for example, three ports hook up to the second processor in the system, and the fourth port hooks up an input-output component such as a chip set. A first signal pathway switching device 1008 such a multiplexer receives a 128-bit packet of information from the link layer. The first signal pathway switching device 1008 reads

the 128-bit packet of information from either the response queue **1011** or the request queue **1010** at a clock speed of 100 megahertz for example. The signal output of the first signal pathway switching device **1008** is two 64-bit packets of information, a first 64-bit packet and a second 64-bit packet which are stored in the first register **1012**. After a time delay **1014**, the second signal pathway switching device **1018** reads the first 64-bit packet through the second register **1016** at twice the clock speed, 200 megahertz, of the 128-bit packet of information. The signal output of the second signal pathway switching device **1018** is two 32-bit packets of information, a first 32-bit packet and a second 32-bit packet which are stored in the third register **1020**. The clock speed of 32-bit packet of information in the third register is again doubled to 400 megahertz.

[0042] The arbiter **1002** now directs the 32-bit packet of information to take a 32-bit bypass path **1022** or a 16-bit information transfer path in the corresponding signal pathway switching devices **1026**, **1034**. The configuration register **1024** informs the arbiter **1002** whether the component exterior to the processor should be a 16-bit, 32-bit or 48-bit-point-to-point connection. In one embodiment, for example, if a 16-bit point-to-point connection exists between the processor and the input-output component, then arbiter **1002** directs each physical layer to act independently of the other physical layers. A fourth signal pathway switching device **1026** reads the 32-bit packet of information at 400 megahertz from the third register **1020** and outputs two 16-bit packets of information. A fifth signal pathway switching device **1028** reads the 16-bit packet of information at 400 megahertz from the forth register **1027** and outputs the 16-bit packet of information to the first port **1004**. The arbiter **1002** sends a signal to the sixth signal pathway switching device **1030** in the second outbound physical layer **1032**. The signal enables the 16-bit path in the fifth signal pathway switching device **1028**. The arbiter **1002**

disables the 32-bit bypass path **1022** from the fourth signal pathway switching device **1026**. The arbiter **1002** also sends a signal to the second physical layer **1032**, the sixth signal pathway switching device **1030**, to enable the 16-bit path in the sixth signal pathway switching device **1030** and disable/ignore the 32-bit bypass path **1022** coming from the third register **1020**. Thus, the output from first port **1004** and the second port **1006** are unrelated 16-bit packets of information.

[0043] If in one embodiment for example, a 32-bit point-to-point connection exist between the processor and the input-output component, then the arbiter **1002** gangs the output of the first port **1004** and the output of the second port **1006**. The arbiter **1002** sends a signal to the sixth signal pathway switching device **1030**. The arbiter's signal disables the 16-bit path in the sixth signal pathway switching device **1030** and enables the 32-bit bypass path **1022** coming from the third register **1020**. The arbiter **1002** also sends a signal to the first physical layer **1031**, the fourth signal pathway switching device **1026**, to enable the 32-bit bypass path **1022**. Thus, the output from first port **1004** and the second port **1006** are related 16-bit packets of information. In other words, the output from first port **1004** and the second port **1006** is a 32-bit packet of information. In order to keep the same transfer speeds, the arbiter **1002** also sends a signal to double the clocking speed that occurs in the registers. For example, the 32-bit packet of information in the third register **1020** is now clocked at 800 megahertz. 800 megahertz is twice the clock speed of when the 16-bit information transfer was occurring. In a similar manner, the arbiter **1002** can gang three ports together (not shown) to create a 48-bit connection in the outbound path.

[0044] In alternative embodiment, if three physical layers are linked to, for example, a second processor, then optionally the arbiter can be programmed to route the three related 16-bit packets

